**OSIS / module**

OSIS information  
and gotchas



**Information and gotchas**

**Contents**

[1 Introduction 3](#_Toc168045453)

[2 A note on OSIS compliance 3](#_Toc168045454)

[3 Detail 4](#_Toc168045455)

[3.1 Non-compliance 4](#_Toc168045456)

[3.1.1 Enclosing tags [NON-COMPLIANCE] 4](#_Toc168045457)

[3.2 Loss of semantics 4](#_Toc168045458)

[3.2.1 Speaker tags [LOSS OF SEMANTICS] 4](#_Toc168045459)

[3.2.2 Acrostic tags [LOSS OF SEMANTICS] 4](#_Toc168045460)

[3.2.3 Selah tags [LOSS OF SEMANTICS] 4](#_Toc168045461)

[3.2.4 Canonical titles at the ends of psalms [LOSS OF SEMANTICS] 4](#_Toc168045462)

[3.3 Arbitrary tweaks with no implications for compliance or semantics 4](#_Toc168045463)

[3.3.1 Avoiding suppression of similar verses 4](#_Toc168045464)

[3.3.2 Notes – functionality 5](#_Toc168045465)

[3.3.3 Notes – rendering of callouts 5](#_Toc168045466)

[3.3.4 Loss of poetry lines 5](#_Toc168045467)

[3.3.5 Blank lines 6](#_Toc168045468)

# Introduction

The title of this document is perhaps slightly misleading. There are a number of issues which can cause problems with a rendered module and which are not obvious from the available documentation. These may lie with OSIS, and the document title implies; but equally they may be issues with osis2mod (either our version or Crosswire’s), or with JSword (again either our version or Crosswire’s), or with our rendering.

Whatever their origin, however, the only place where we can readily influence matters is in the OSIS which we generate and pass to osis2mod.

# A note on OSIS compliance

In theory, Crosswire require that modules are fully OSIS-compliant. By this, they mean compliant not with the OSIS documentation (which is not necessarily normative, and is in any case difficult to interpret in some cases) but with the OSIS XSD. Unfortunately there seems to be no official version of this; I was pointed at someone’s private copy, which is itself no longer available online (although I do have a copy). However, even compliance with this seems to be no absolute guarantee that modules will be handled completely correctly by osis2mod / JSword / STEPBible. More details of some of the issues appear in section 3.

Note, anyway, that the modules we generate *aren’t* fully compliant. I highlight the deviations in section 3.

# Detail

The following sections are in no particular order.

Some of the information below implies that we are deviating from OSIS compliance. Some implies that, although compliant, I am having to sacrifice semantic details in order to make things work. Both of these I highlight. Others are simply odd ways of working which appear to be required, but which have no further implications.

There are also some more complex issues which are discussed in sections 4 onwards.

## General things to be aware of

### Span-type tags and verse markers

According to the OSIS XSD, things like <hi> are not permitted to contain verse markers. There is some anecdotal evidence to suggest that breaking this particular rule can indeed give rise to problems (see section 4). I suspect there may be similar issues with other span-type tags.

### Indented paragraphs

Translators commonly use poetry or list tags on the assumption that this will generate indented paragraphs. In fact it does not do so. My guess here – although it is *only* a guess – is that osis2mod turns these tags (which *enclose* their content) into milestone tags (which precede what would otherwise be their content).

I think it does this to avoid cross-boundary markup. However, while an enclosing tag can apply a left and right margin to its content, a milestone tag cannot. Hence at best this can indent the first line.

## Non-compliance

### Enclosing tags [NON-COMPLIANCE]

OSIS requires that poetry and list elements be enclosed in a tag equivalent to HTML’s <ul> tag.

This is problematic, both in that it tends to give rise to cross-verse-boundary markup (which is a problem for reversification and may also be an issue for osis2mod), and in that if retained, it gives rise to excessive vertical whitespace when rendered.

We therefore don’t bother with these tags – things seem to cope fine without, and the rendered output looks better. (In fact I have a feeling that even if we retained them, osis2mod may itself remove them.)

### Copyright information [NON-COMPLIANCE]

Crosswire support a very limited form of markdown to be used on copyright and related information – there are very few tags, and the ones that there are can be used only in selected fields in the Sword configuration file.

In fact, it appears that many people ignore this, and use HTML instead, and we do this too – it makes it possible to improve the readability of the copyright information, and it also means we can simply pick up the information directly from DBL configuration files when working with DBL data (the metadata for quite a lot of texts contains relevant fields in HTML form).

## Loss of semantics

### Speaker tags [LOSS OF SEMANTICS]

Speaker tags do work, but we don’t like the way they are formatted, so I am replacing them by formatting markup.

### Acrostic tags [LOSS OF SEMANTICS]

Acrostic tags (as both paragraphs and as span-type) tags do work, but we don’t like the way they are formatted, so I am replacing them by formatting markup.

### Selah tags [LOSS OF SEMANTICS]

In theory, selah tags are supposed to be rendered on the current line, right justified. This doesn’t happen, so I am replacing them by an alternative formatting markup.

### Canonical titles at the ends of psalms [LOSS OF SEMANTICS]

There are a few places where psalms have canonical titles at their ends. If left as such, weird things happen (eg verse numbers being misplaced or verses being moved to the next chapter). I therefore convert them to formatting markup.

### Notes and cross-references functionality [LOSS OF SEMANTICS]

Footnotes and cross-references do not appear to work outside of verses.

* The *callout* for the note or cross-reference appears, but nothing happens if you hover the mouse over it.
* In some cases you can remedy this by moving the note tag within an adjacent verse (so long as you can decide whether the note belongs with the previous verse or the next one).
* However, there are some cases where this won’t work. For example, I believe I am correct in saying that a note marker within a canonical header is treated as being outside of a verse, and therefore does not work – and you are unlikely to have any obvious place within the first verse to which you can move the note.
* We have no experience of footnotes within introductory material, because at the time of writing we can’t render introductory material.

## Arbitrary tweaks with no implications for compliance or semantics

### Avoiding suppression of similar verses

If you have two or more consecutive verses which are similar (identical?), something somewhere suppresses some of them. You might assume this is an unlikely eventuality, but it is actually quite likely to occur, particularly when expanding elisions, because here we end up with a number of consecutive verses all of which contain nothing more than a dash (our attempt at indicating the verses are not empty by accident).

You can get round this by enclosing the content of alternate verses in an otherwise entirely pointless  
<hi type='normal'>.

### Notes – rendering of callouts

This is slightly confused …

Sometimes the text has a number of consecutive notes / cross-references and separates them by commas – and the commas are randomly rendered or not.

Sometimes the text has a number of consecutive notes / cross-references and does not separate them by commas – but commas are inserted by something before we get as far as the rendered output.

It is not entirely clear what effect we want to achieve. At present, all callouts appear as down-arrows, and where this is the case, there seems to be little point in them being separated by commas. Should we ever get to the point where we use anything else as callouts, the commas might be useful to aid readability.

Anyway, if you have commas and wish to retain them, adding an empty <hi type='normal'> before the note tags seems to do the trick.

And if you do *not* have commas and wish to prevent them from being added spontaneously, adding a single space between the note tags seems to work. (That’s a plain vanilla space – ASCII 0x20. You don’t need a non-breaking space or anything like that.)

### Loss of poetry lines

On at least one text recently, poetry lines were being dropped in an arbitrary manner (ie some poetry lines appeared in the output while others did not).

To address this I am inserting an otherwise entirely pointless <hi type='normal'> before any poetry tag.

### Blank lines

It has latterly become apparent that something somewhere can get screwed up if blank lines appear in the text in certain places.

Blank lines at the very end of a chapter cause the last verse number to come out after the text of the verse so I suppress these.

I have also noticed in at least one case that a blank line actually caused a later verse to be dropped in its entirely (and not even an adjacent verse). This is clearly worrying, but at present I have no real handle on what is going on.

Experience also suggests that a para:b (<lb>) introduced not to split an existing line but purely to force a blank line may be ignored. If you really want the blank line to appear, the line actually needs to have something on it (like &nbsp;).

# Tables

Tables which contain verse markers are a particular problem (except possibly if, throughout the table, the sid and eid for each verse are both in the same cell, but I have yet to encounter a table like that).

Any attempt at all to retain tables which contain verse markers is likely to result in *osis2mod* complaining or – worse – there being no complaint, but things coming out entirely wrong. And of course verses which have markup running across their boundaries are a big problem for reversification (the one redeeming feature here being that I believe it unlikely that those portions of a text most likely to be subject to reversification will be formatted as tables).

To address this, I retain the table markup, but remove the verse markup, creating an elision instead. Thus if the table originally spanned vv1-10, we end up with vv2-9 empty, and the entire table in v1. This retains the tabular appearance, but at the cost of the individual verses no longer having their original content (with unavoidable knock-on implications for added value such as verse vocabulary, interlinear, etc).

As regards table tags, the OSIS reference manual admits that support for tables is somewhat half-hearted, and also suggests that implementations will need to add their own ‘*x-*’ attributes in support of processing (something which surely of itself limits the usefulness of making OSIS modules available to third parties). STEP does precisely this, with attributes like *x-simpleTable*, for instance. Unfortunately, though, I have been unable to find any documentation as to what attributes STEP requires or supports. I can only say that with the text I have handled to date, we have successfully handled *style='tr'* on USX *row* tags, and things like *<cell align='start' style='tc1'>* and *<cell align='end' style='tcr2'>* on USX cells – so the OSIS we are generating for these is clearly appropriate. The USX *align* parameter presumably dictates alignment (and is actually the one standard attribute defined in the manual), so I am not sure why ‘r’ is needed in the second example, since presumably it, too, refers to right alignment. I have a feeling there are quite a number of other possibilities, but I have no idea what they are.

STOP PRESS: We have recently come across an alternative formalism for tables, in which table- and row- tags are not used. With a two-column table, the content of each entry in the left-hand column is enclosed in <cell>, and the right hand column follows as plain vanilla text. This may work only with two-column tables (need to investigate), and to my mind produces something which is not well enough formatted to be useful, but we’ll have to see.

# Canonical titles at the start of Psalms [POSSIBLE LOSS OF SEMANTICS]

Canonical titles may appear both at the start and at the end of certain psalms. Both give rise to issues. The titles which appear at the *ends* of psalms are discussed in section 3.3.4. The present section discusses titles which appear at the *starts* of psalms. These give rise to much more significant issues.

Texts may be handled in any one of three different ways – they may be processed more or less as-is, using Crosswire’s osis2mod and one of the osis2mod versification schemes; they may be restructured to NRSV form using the reversification data during the conversion (in which case it is open to us to use Crosswire’s osis2mod and their NRSV scheme); or they may be samified.

Each of these gives rise to its own issues, and each is dealt with in its own section below.

Orthogonal to this choice of osis2mod and output versification scheme, there is also the matter of the versification tradition which the raw text uses – does it, for instance, basically follow the Hebrew scheme, the Latin scheme or what.

And orthogonal to *that* is the matter of markup. Different raw-text versification schemes probably, I guess, imply different approaches to markup; but in any case, I imagine different translators are going to do things differently.

And having dealt with three lots of orthogonal issues, I now have to move into the fourth dimension to cater for the fact that regardless of what we do, there are certain things which just don’t seem to work well with osis2mod, so we need to cater for that too.

At the time of writing, we are working particularly with GerHFA, so what follows is really conditioned by the issues we are encountering with that (and means that at present we are looking at only one raw-text versification scheme). It would be very nice if the issues and solutions identified here would transfer to other forms of text too (or at least, not flatly contradict them), but that remains to be seen.

## General points

### Replacing psalm:title [LOSS OF SEMANTICS]

It looks as though in some situations (and perhaps all) we shall have to replace title:psalm by formatting markup – most likely hi:italic. Retaining the psalm:title, at least with GerHFA, was giving empty verses, verse numbers in the wrong place, etc.

Note that hi:italic is not rendered in quite the same way as psalm:title. The latter is rendered using a sans serif font, whereas hi:italic simply italicises body text (which uses a serif font).

Note also that psalm:title is automatically set off on a line of its own, so that it does indeed look like a header. hi:italic is not, so a newline needs to be introduced after it (but ideally not if the text already has one there, or there will be excessive vertical whitespace).

### Reversification footnotes

Reversification processing may call for footnotes to be added to canonical titles. It is necessary to suppress these notes, because, as discussed in section 3.3.5, footnotes outside of verses don’t function. Regardless of whether we use title:psalm or hi:italic, the canonical text is treated by osis2mod et al as though it *is* outside of a verse.

## Using Crosswire’s osis2mod with one of their schemes

I had hoped here that we might simply be able to run with the general structure of the input text. Sadly, this turns out not to be the case.

Different psalms give rise to different kinds of issues, so I shall need to look at each separately.

To give a stake in the ground, GerHFA is well aligned with the Crosswire German versification scheme, which is the one I have chosen to use (it is also well aligned with Luther – I suspect the two are pretty much identical). Or more accurately, the books, the numbers of chapters per book, and the numbers of verses per chapter fit well. I do not consider canonical titles in making this assessment, because I lack details of any expectations Crosswire may have in that respect.

In the following examples, I give a simplified version of the markup which would be derived directly from the GerHFA USX, and then discuss how this needs to be altered (assuming it does) in order to look ok in the rendered output. My aim, as far as possible, has been to align it with the text as it appears at bible.com.

### Ps 3

**<title><verseStart:1/> contents … <verseEnd:1/></title>**

According to the OSIS XSD, title:psalm can contain a verse. In practice, this doesn’t seem to work: indeed retaining the title:psalm at all gives rise to problems.

This therefore needs to be turned into:

**<verseStart:1/><hi:italic> contents … <hi:italic/><verseEnd:1/>\*\***

An additional line break or two may be needed at \*\*. See section 5.1.1 for a discussion of the ordering of the tags here.

### Ps 11

**<title><verseStart:1/>TitleText</title> VerseContents … <verseEnd:1/>**

This needs to turn into

**<verseStart:1/><hi:italic>TitleText</hi:italic/>\*\* VerseContents … <verseEnd:1/>**

An additional line break or two may be needed at \*\*. See section 5.1.1 for a discussion of the ordering of the tags here.

### Ps 51

**<title><verseStart:1/> VerseOneContents … <verseEnd:1/></title>**

**<title><verseStart:2/> VerseTwoContents … <verseEnd:2/></title>**

This needs to turn into

**<verseStart:1/><hi type='italic'> VerseOneContents … </hi><verseEnd:1/>\***

**<verseStart:2/><hi type='italic'> VerseTwoContents … </hi><verseEnd:2/>\*\***

In the GerHFA verse 2 has a footnote. In this particular case (contra section 5.1.2) the footnote does not need to be suppressed, because it will still fall within a verse after processing.

An additional line break may optionally be added at \* (assuming one is not there already) if it is desired to split the dual verse title on to two lines.

And a further line break may be needed at \*\* (again if not already present) to force the title to be set off from the following text.

## Conversion-time restructuring

As a reminder, this entails using the reversification data during the conversion process to restructure the text so as to make it NRSV-compliant. Following this we should be able to use the Crosswire version of osis2mod to create a module (we *could* use our own version, but there is probably little point, since there is nothing for it to do).

I am finding it difficult to work out what is required here. Previously I have been driving any reformatting directly from the reversification data, but I am not sure that’s enough now. Based upon what we have seen with GerHFA, it looks as though we may need to be driven at least in part by the sort of considerations set out in section 5.2.

In other words, we need something which not only fits with NRSV, but which also fits with any idiosyncrasies of osis2mod’s processing (eg not placing verses inside hi:italic).

To summarise some of the issues …

I need to be able to identify which reversification rows apply. Some of those involve checking whether there is an existing title. We potentially have two different ways of doing this – we can look for existing psalm:title markup as we did in section 5.2; or, following more recent discussions, we ignore any existing markup and simply look for canonical text which appears prior to the start of v1, regardless of how that canonical text is marked.

Note that if we were to apply that latter test to the text of Ps 3, 11 and 51 as shown in section 5.2, none of those chapters would show up as having existing headings. I am not sure whether the reversification data is set up to work on that basis, or whether it expects the existing headings to be identified.

**\* End of document \***